

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matevž Celarc

**Avtomatizacija krmiljenja toplotne
tehnike na kopališču**

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJ RAČUNALNIŠTVA IN
INFORMATIKE

MENTOR: izr. prof. dr. Marko Bajec

Ljubljana 2013

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.¹

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

¹V dogovoru z mentorjem lahko kandidat diplomsko delo s pripadajočo izvorno kodo izda tudi pod katero izmed alternativnih licenc, ki ponuja določen del pravic vsem: npr. Creative Commons, GNU GPL.



Št. naloge: 01924/2013

Datum: 02.04.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **MATEVŽ CELARC**

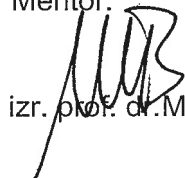
Naslov: **AVTOMATIZACIJA KRMILJENJA TOPLOTNE TEHNIKE NA
KOPALIŠČU
AUTOMATION OF THE SWIMMING POOL THERMAL CONTROL**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

V okviru diplomske naloge izdelajte sistem za avtomatizacijo krmiljenja ogrevalne tehnike javnega bazena. Preštudirajte in predstavite koncept centralnega sistema nadzora in opišite vso potrebno toplotno tehniko in strojno opremo. Izdelajte načrt potrebne programske opreme za krmiljenje ter implementirajte strežnik.

Mentor:



izr. prof. dr. Marko Bajec

Dekan:



prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Matevž Celarc, z vpisno številko **63080044**, sem avtor diplomskega dela z naslovom:

Automatizacija krmiljenja toplotne tehnike na kopališču

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom izr. prof. dr. Marka Bajca,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 17. maja 2013

Podpis avtorja:

*Zahvalil bi se vsem, ki so pripomogli k nastanku diplomske naloge.
Posebna zahvala pa gre mentorju, sošolcem in Jasni.*

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Centralni sistem nadzora	3
2.1	Opis sistema	4
2.2	Struktura sistema	5
2.3	Razlog razvoja	7
2.4	Odločitvena logika avtomatike	7
2.5	Shramba podatkov	8
2.6	Oddaljeni nadzor	8
3	Strojna oprema	11
3.1	Toplotna tehnika	11
3.1.1	Črpalke	11
3.1.2	Ogrevalne naprave	12
3.2	Temperaturni senzorji	13
3.2.1	IP modul 1W340A	13
3.2.2	Senzor DS18B20	13
3.2.3	USB senzor TEMPer1	14
3.3	Releji	14
3.3.1	Denkovi DAEnetIP2	14

4	Razvoj sistema	17
4.1	Orodja, protokoli in tehnologije uporabljene pri razvoju	17
4.1.1	Tehnologije	17
4.1.1.1	Spletne storitve	17
4.1.1.2	Mobilni operacijski sistem Android	19
4.1.1.3	SQLite	20
4.1.2	Protokoli	20
4.1.2.1	SNMP	20
4.1.3	Razvojna orodja	21
4.1.3.1	Visual studio 2010	22
4.1.3.2	Eclipse in ADT vtičnik	22
4.1.3.3	Wireshark	23
4.2	Strežnik	24
4.2.1	Opis zgradbe aplikacije	24
4.2.2	Gonilniki za nadzor strojne opreme	25
4.2.2.1	Zajem temperaturnih vrednosti	27
4.2.2.2	Nadzor rele modulov	29
4.2.3	Shranjevanje podatkov	29
4.2.4	Implementacija odločitvene logike	30
4.2.5	Oddaljeni nadzor	31
4.2.5.1	Zagotavljanje varnosti	33
4.3	Odjemalci	35
4.3.1	Opis splošnih funkcionalnosti	35
4.3.2	Windows aplikacija	35
4.3.2.1	Prilagoditve vmesnika	36
4.3.2.2	Opis funkcionalnosti	37
4.3.3	Mobilna aplikacija Android	39
4.3.3.1	Splošno o mobilnih aplikacijah	40
4.3.3.2	Opis funkcionalnosti	41
5	Sklepne ugotovitve	43

Povzetek

V diplomskem delu je predstavljen model avtomatizacije krmiljenja toplotne tehnike za konkretni primer kopališča Laguna Ljubljana. Namen diplomske naloge je bil zajeti specifične zahteve in implementirati sistem, ki bo v danih razmerah deloval optimalno.

V prvem delu je opisan koncept centralnega sistema nadzora in kako se ta navezuje na konkretni primer kopališča. Predstavljeno je stanje pred uporabo novega sistema z vso pripadajočo toplotno tehniko in strojno opremo. Nato sledi podroben opis naprav nameščenih v nov sistem. V drugem delu diplomske naloge smo se osredotočili na implementacijo sistema in notranjo zgradbo strežniške aplikacije. Natančno je opisana večina pomembnih strežniških komponent. Posebej smo se osredotočili na analizo zaprtih protokolov, ki so uporabljeni za komunikacijo s strojno opremo. Pri analizi smo uporabili posebna orodja in obratno inženirstvo. V zaključku pa smo predstavili različne odjemalce storitev in s tem tudi možnosti, ki jih nudi razvit sistem.

Sistem je v uporabi na kopališču Laguna Ljubljana že od septembra 2012.

Ključne besede

centralni sistem nadzora, avtomatsko upravljanje, nadzor na daljavo, senzor-sko omrežje, senzorji, rele enote

Abstract

The thesis presents a model of thermal control automation techniques for the case of Laguna Ljubljana resort. The aim of the thesis was to get specific requirements and implement a system that will work optimally in a given situation.

First section describes the concept of a central control system and how it relates to real life usage in resort. State prior to use of the new system is described with all thermal equipment and hardware. This is followed by a detailed description of the devices, which were used in the new system. Second section is focused on the implementation of the system and the internal structure of server application. Most server components are described in details. We focused on the analysis of the proprietary protocols that are used to communicate with the hardware. For protocol analysis we used specialized tools and method of reverse engineering. At the end, we look all client applications used to control system and their features.

The system is used on Laguna Ljubljana resort since September 2012.

Keywords

central control system, automatic control, remote control, sensor networks, sensors, relays

Poglavje 1

Uvod

V diplomskem delu je predstavljena izvedba avtomatizacije toplotne tehnike, natančneje avtomatizacija krmiljenja toplotne tehnike na podlagi podatkov senzorske mreže. Osnovni namen avtomatizacije je zagotavljanje optimalnega izkoristka toplotne energije in občutnega zmanjšanja oziroma odprave potrebe po ročnem upravljanju. Diplomsko delo je bilo načrtovano po meri za toplotni sistem kopališča Laguna Ljubljana.

Topotni sistem na kopališču je zaprt krožni tok toplote. Nanj so priključeni različni elementi, ki toplotno energijo bodisi dovajajo ali izkoriščajo. Elementi so med sabo povezani preko visokotlačnih cevi, v katerih se za transport toplote uporablja voda. Dovod toplote do posameznega elementa uravnavamo preko črpalk, katere uravnavajo tok vode v zaprtem krogu. Stari sistem je bil v veliki meri krmiljen ročno. Sicer so bili nekateri deli sistema avtomatizirani, vendar samo do neke mere. Vsi ti različni sistemi pa med sabo niso znali komunicirati. Tak način delovanja je zelo neučinkovit in izpostavljen napakam. Izvedba avtomatizacije vključuje združitev vseh sistemov na enoten način delovanja.

Avtomatika nadzora za svoje delovanje uporablja vrednosti pridobljene iz mreže temperaturnih senzorjev. Temperaturni senzorji so nameščeni na strateških mestih v toplotnem krogu. Na podlagi uporabniško določenih pogojev se vrednosti senzorjev uporabi za vklop ali izklop ustrezne črpalke.

Posamezno črpalko krmilimo preko rele enot. Sistem omogoča centralni nadzor celotnega sistema in med drugim tudi vpogled v zgodovino dogajanja v sistemu. Največja prednost novega sistema je tudi možnost nadzora na daljavo preko različnih mobilnih naprav.

Poglavje 2

Centralni sistem nadzora

Vsak posameznik si želi, da bi lahko nadzoroval in upravljal dogajanje v svoji hiši, ne glede na razdaljo. V ta namen se je razvil koncept centralnega sistema nadzora, ki se je začel uporabljati za nadzor elementov v hiši. Kmalu se je razvil pojem "pametna hiša", ki omogoča uporabniku popoln nadzor nad hišo ne glede na oddaljenost od nje. Ideja pametne hiše se je nato pokazala kot edinstvena rešitev za podjetja, ki imajo v lasti poslovne stavbe. Podjetjem je zelo pomembno, da imajo učinkovit nadzor nad ogrevanjem in hlajenjem prostorov, ki so v njihovi lasti.

Hiter razvoj tehnologije pa nam dandanes že omogoča združitev vseh naprav, ki so zadolžene za uravnavanje temperature v enoten sistem. Najpomembnejša lastnost takega sistema je avtomatski nadzor, kar pomeni popolnoma samodejno delovanje naprav. Avtomatski nadzor poskrbi za ustrezno delovanje sistema glede na pogoje, ki jih določimo. Tvrsten sistem je zelo učinkovit, saj v veliki meri odpravi možnost človeških napak.

Poleg nadzora pa mora sistem beležiti tudi vse podatke o delovanju naprav in temperaturnih vrednostih, saj to omogoča uporabniku natančen pregled nad delovanjem sistema. V primeru, da se zgodi okvara določene naprave ali pa se pojavijo nepravilne temperaturne vrednosti, se napako lahko hitro odkrije prav zaradi beleženja podatkov in prikaza trenutnega stanja. Tako je možno odkrite napake odpraviti v najkrajšem možnem času. Pomemben

del sistema za nadzor je seveda ustrezna aplikacija, ki uporabniku omogoča enostaven nadzor sistema na daljavo.

Uporaba računalnikov, mobilnih in tabličnih računalnikov se je dandanes razširila že do te mere, da si ljudje ne predstavljajo življenja brez njih. Ravno zaradi hitrega razvoja tehnologije se je znatno povečalo tudi povpraševanje po nadzoru objektov na daljavo. Centralni nadzorni sistem tako podjetjem omogoča, da imajo vedno na voljo:

- trenutno stanje temperaturnih vrednosti,
- celotno zgodovino stanja,
- nastavitve avtomatskega nadzora,
- možnost ročnega nadzora.

Centralni sistem nadzora predstavlja podjetjem optimalno rešitev za uravnavanje temperature v objektih, saj omogoča avtomatizacijo na področju zagona naprav, nadzora temperaturnih vrednosti in beleženju informacij. Za podjetja je uporaba takega sistema optimalna, ker jim nudi:

- hitro in enostavno pridobitev informacij o stanju sistema;
- odsotnost človeškega faktorja, kar zmanjša možnost napak;
- zmanjševanje stroškov.

2.1 Opis sistema

Bistvo dobrega centralno nadzornega sistema je avtomatski nadzor, ki zmanjša vlogo človeškega faktorja. Glavna prednost avtomatskega nadzora je samostojno delovanje, saj uveljavlja spremembe glede na zunanje pogoje. Sistemu je potrebno določiti pogoje, ki bodo opredelili, kdaj naj se spremembe izvedejo. Pogoji so lahko določeni glede na vrednost ali časovni interval. Ko

so pogoji na sprožilcu izpolnjeni, se izvršijo določene procedure. Najpomembnejša med ukazi, ki jih omogoča sistem in ki uravnavata temperaturo v objektih, sta vklop ali izklop naprave.

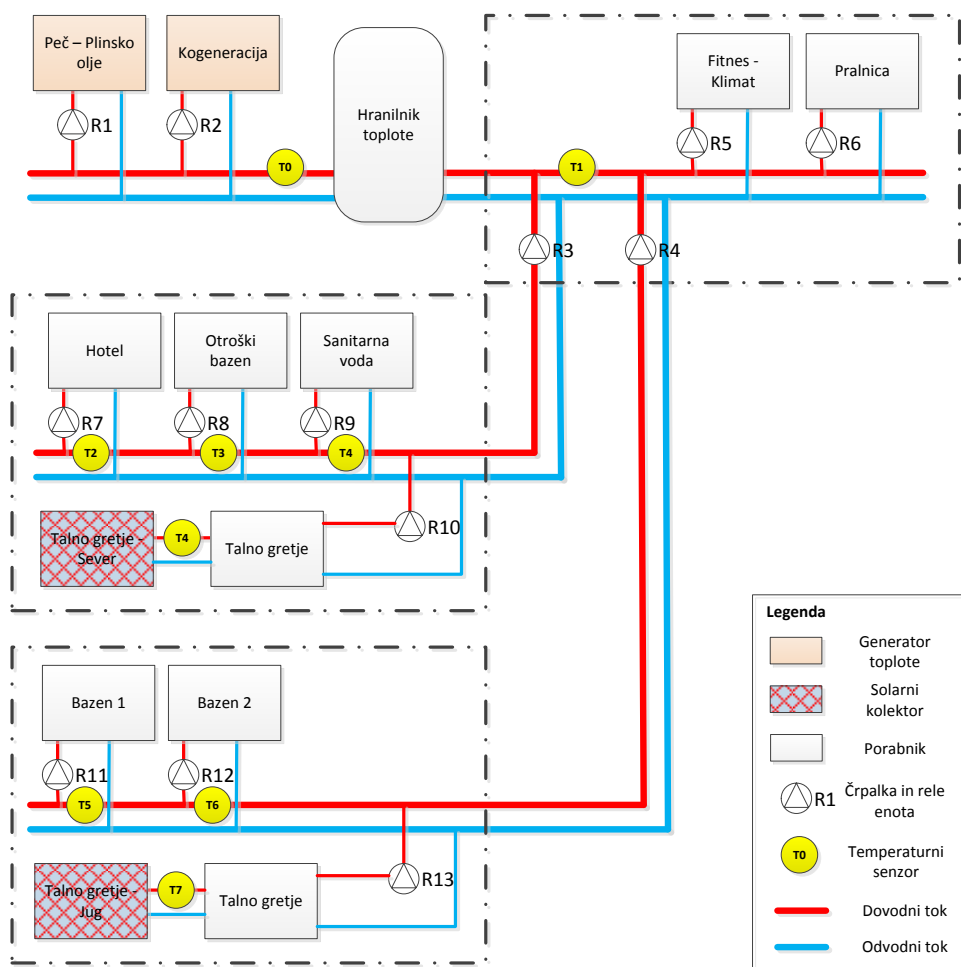
Poleg avtomatskega nadzora pa sistem omogoča tudi ročni nadzor. V primeru, da želimo izvršiti določene kratkotrajne spremembe temperature, lahko vključimo ročni način, s katerim lahko nato vklopimo ali izklopimo določene naprave v sistemu.

Prednost centralno nadzornega sistema je nadzor na daljavo. Ta nam omogoča pregled nad delovanjem sistema in stanjem temperature ne glede na to, kje se nahajamo. Sistem nam omogoča vpogled v podatke o temperaturnih vrednostih v obliki grafa, ki se izriše glede na izbrano časovno obdobje. To je predvsem pomembno v primeru, kadar podjetje želi opraviti natančen pregled in analizo podatkov za preteklo obdobje.

2.2 Struktura sistema

Toplotni sistem na kopališču Laguna Ljubljana je zaprt toplotni krog. Poenostavljeno shemo sistema s priključenimi napravami vidimo na sliki 2.1. V legendi so označeni različni tipi naprav, ki so uporabljene v sistemu. Vsaka priključena toplotna naprava je v sistem povezana z dovodnim in odvodnim tokom. Posamezni deli sistema se nahajajo v različnih stavbah in so med sabo lahko oddaljeni tudi do 300m.

Za prenos toplote se uporablja namenska voda, ki nikoli ne zapusti sistema. Toplo ali hladno vodo pa znotraj sistema potiskajo namenske črpalke. Na vsaki črpalci je nameščena rele enota, s katero krmilimo črpalko. Na ključnih mestih sistema se nahajajo temperaturni senzorji. Temperaturni senzorji so praviloma nameščeni na dovodni tok in blizu črpalk. Ta postavitev nam omogoča, da posameznemu delu sistema ali napravi dovedemo toploto, samo ko je to potrebno.



Slika 2.1: Shema sistema na kopališču Laguna Ljubljana

2.3 Razlog razvoja

Razvoj osebnih računalnikov, interneta in drugih tehnologij je privedel ljudi do razmišljanja o razvoju sistema, ki bi poskrbel za samostojno delovanje naprav. Podjetja so vedno bolj zainteresirana za uporabo tovrstnega sistema, ker jim to omogoča zmanjšanje stroškov in predvsem olajša upravljanje objektov.

Dandanes ima skoraj vsaka kupljena naprava možnost oddaljenega nadzora. Med te naprave spada oprema, kot so temperaturni senzorji in rele enote. Za podjetja uporaba teh možnosti predstavlja problem, ker nadzor naprave poteka preko več različnih programskih rešitev, ki med sabo niso kompatibilne. Velikokrat pa se zgodi, da nadzor vseh naprav ne deluje na osnovi enakega koncepta.

Glavni razlog za razvoj sistema je bil izražen že s strani podjetja, ker so želeli enoten sistem, ki omogoča avtomatiko nad upravljanjem s temperaturo, kar jim možnosti na kupljeni opremi niso omogočale. Naslednji problem, ki se je pojavil pri uporabi nadzora prisotnega na strojni opremi, je ročno vklopjanje in izklopjanje opreme. Pri takem načinu delovanja se povečajo možnosti človeške napake, avtomatski nadzor pa možnost napak v veliki meri odpravi.

Uporaba pametnih telefonov in tabličnih računalnikov se povečuje iz dneva v dan. To je bil poglobitveni razlog za razvoj sistema, ki bo omogočal enostaven nadzor na daljavo, tudi preko telefona ali tabličnih računalnikov.

2.4 Odločitvena logika avtomatike

Avtomatski način nadzora deluje glede na pogoje, ki so določeni znotraj definiranih sprožilcev. Pogoji v sistemu so lahko določeni glede na vrednost ali čas. Vrednostni pogoj je lahko določen glede na senzorje ali releje. Prva možnost za določitev pogoja je primerjava med definirano vrednostjo in vrednostjo pridobljeno iz senzorja. V našem primeru lahko določimo, katero temperaturo mora senzor imeti, da pride do spremembe v sistemu. Drugi

način uporabe vrednostnega pogoja je osnovan na stanju releja. V pogoju bi lahko revidirali, če je rele vklopljen ali izklopljen. Časovni pogoj je primerjanje trenutnega časa s časovnim intervalom, ki ga določimo. Ko pride do različnih časovnih postavk, je pogoj izpolnjen in sprememba v sistemu se zgodi.

2.5 Shramba podatkov

Sistem od pričetka svojega delovanja shranjuje vse pomembne podatke. Med najosnovnejše podatke, ki se morajo shranjevati, spadata temperatura in čas njihove meritve. Podatki se beležijo glede na vrednost in čas, v katerem jih je senzor izmeril.

Poleg temperature pa je pomembno shranjevati tudi podatke o sprožilcih. Ti podatki nam omogočajo pregled nad njihovim delovanjem v izbranem časovnem obdobju. Sistem nam prikaže podatke o sprožilcih, ki so vključeni v sistem ter o njihovem statusu in času, ko je bil ta spremenjen. Poleg osnovnih podatkov nam sistem izpiše tudi pogoje, ki so prisotni na sprožilcu. Sistem shranjuje tudi vse spremembe, ki se zgodijo na rele enotah. Prav tako kot temperaturne vrednosti in sprožilce lahko tudi podatke o relejih pregledujemo glede na izbrano časovno obdobje. Poleg izpisa podatkov o relejih, njihovem statusu in času sprememb nam sistem zbrane podatke prikaže tudi v obliki grafa.

2.6 Oddaljeni nadzor

Kot smo že omenili, je najpomembnejša funkcija sistema oddaljeni nadzor. Ta nam prikazuje seznam vseh prostorov, ki so vključeni v sistem in njihove temperaturne vrednosti. Če želimo pridobiti podrobne informacije o stanju temperature, lahko izberemo zelene prostore in časovno obdobje, za katero želimo izpis podatkov. Podatki se nato izpišejo v obliki grafa.

Pomembna informacija, ki jo lahko pridobimo preko oddaljenega nadzora,

je delovanje senzorjev. Če pride do okvare senzorja, nam sistem s spremembo barve senzorja prikaže napako. Poleg temperature in senzorjev nam sistem omogoča tudi nadzor nad rele enotami, ki so vključene v sistem. V primeru, da vklopimo možnost ročnega nadzora sistema, nam ta omogoči vklapljanje in izklapljanje relejev. Spreminjanje njihove vrednosti pa ne bo delovalo, dokler je vklopljen avtomatski način delovanja.

Preko oddaljenega nadzora pa lahko pregledujemo tudi vso zgodovino vrednosti in dogodkov. Ta se beleži od prve uporabe sistema in nam tako omogoča, da preverjamo njegovo delovanje v določenem časovnem obdobju.

Poglavje 3

Strojna oprema

V tem poglavju je predstavljena strojna oprema, ki je potrebna za ustrezno delovanje sistema.

3.1 Toplotna tehnika

Sistem za svoje delovanje potrebuje naprave, ki dovajajo toploto. Toplota med napravami potuje s pomočjo črpalk. Vse naprave v sistemu so med sabo povezane s cevmi v zaprti toplotni tok. Vsaka naprava v sistemu ima dovodno in odvodno cev.

3.1.1 Črpalke

Sistem za prenos toplote uporablja dvocevne obtočne črpalke. To v praksi pomeni, da imamo dovod vroče vode in odvod iz katerega prihaja ohlajena voda. Najpogostejša črpalka uporabljena v sistemu je GRUNDFOS UPS 50–180 F. Primer črpalke lahko vidimo na sliki 3.1. Črpalka ima mokrotekoči rotor, njeno ohišje je zgrajeno iz litega železa. Temperaturna tekočine lahko doseže od $+ 2^{\circ}\text{C}$ do 110°C [1].



Slika 3.1: GRUNDFOS UPS 50 – 180 F.

3.1.2 Ogrevne naprave

Za dovajanje toplote v sistem se uporabljajo različne naprave. Uporaba posamezne naprave je odvisna od vremenskih pogojev in zahtev po toploti. Možna je tudi souporaba več naprav sočasno.

Kogeneracija Neodvisni sistem za proizvodnjo električne energije kot stranski produkt proizvaja toploto. Oprema znotraj generatorja omogoča, da s tem grejemo vodo v zaprtem sistemu.

Peč na kurilno olje V primeru, da generator električne energije ne proi-

zvede dovolj toplote ali v času ekstremnih vremenskih razmer, se uporabi peč na kurilno olje.

Sončni kolektor Naprava izkoristi sončno toploto in s tem greje vodo v sistemu. Sončni kolektor se uporablja v toplejših mesecih in samo za gretje sanitarnih vod.

3.2 Temperaturni senzorji

Podatke o temperaturah sistem pridobi s pomočjo senzorjev, ki so prisotni v prostorih. Vsi senzorji so priključeni na nadzorno napravo, ki omogoča komunikacijo z njimi.

3.2.1 IP modul 1W340A

Nadzorna oprema uporabljena za temperaturne senzorje v sistemu je 1W340A-D4. Naprava podpira standard 1-Wire [2] in nudi podporo temperaturnima senzorjema DS18B20 in FM75. Oba sta uporabljena v centralnem sistemu nadzora. Na napravo lahko priključimo do 12 senzorjev. Napravo priključimo na ethernet omrežje. Dostop do naprave in podatkov poteka preko TCP/IP protokola. Primer naprave lahko vidimo na sliki 3.2.

3.2.2 Senzor DS18B20

Senzorji DS18B20 so uporabljeni za zajemanje temperaturnih vrednosti v °C. Komunikacija poteka preko 1-Wire standarda, ki omogoča komunikacijo med več napravami z minimalno uporabo nožic na eni izmed računalniških vrat. Ime je dobil zaradi uporabe enega samega vodila za prenos podatkov. Senzorji imajo parazitski način delovanja, kar pomeni, da lahko pridobivajo moč preko podatkovnega vodila in zato ne potrebujejo dodatnega zunanjega napajanja. Senzor lahko meri temperature med -55°C in 125°C z natančnostjo $\pm 0,5^{\circ}\text{C}$ [3]. Vsak senzor DS18B20 ima unikatno 64-bitno serijsko številko, ki omogoča enolično identifikacijo.



Slika 3.2: Temperaurni modul 1W340A-D4

3.2.3 USB senzor TEMPer1

Sistem uporablja poleg omenjenih senzorjev v poglavju 3.2.1 1-wire senzor tudi USB senzor TEMPer1. Naprava je namenjena merjenju temperature preko USB priključka. Naprava podpira merjenje temperature med vrednostmi -40 in $+120$ °C [4]. Eden izmed glavnih razlogov za uporabo omenjenega senzorja je ugodna cena, saj znaša približno 5 EUR. Primer senzorja lahko vidimo na sliki 3.3.

3.3 Releji

Rele enote, ki so prisotne v sistemu so priključene v nadzorno enoto. Število relejev, ki jih je mogoče priklopiti na enoto je odvisno od modela. V našem sistemu uporabljamo nadzorne enote Denkovi DAEnetIP2.

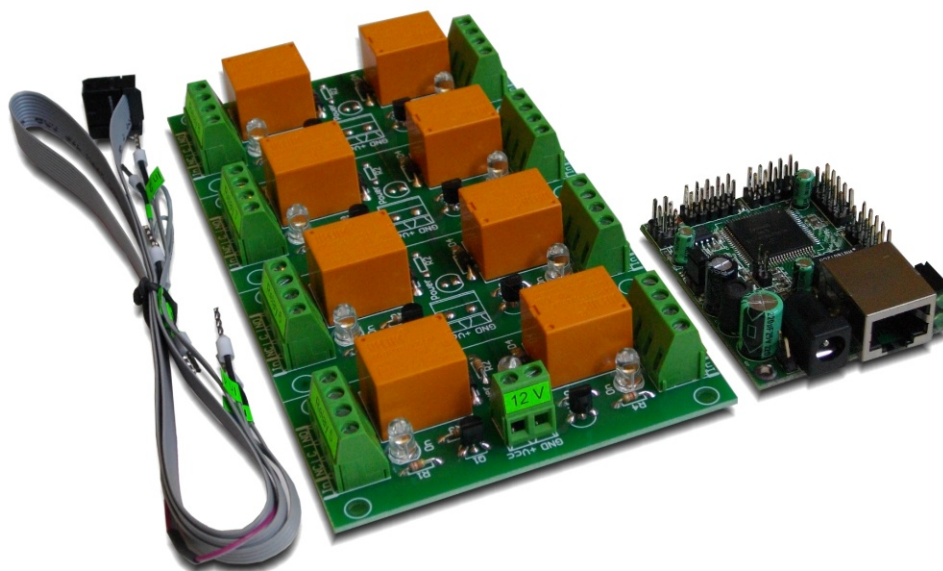
3.3.1 Denkovi DAEnetIP2

Nadzorna enota DAEnetIP2 je namenjena avtomatizaciji doma ali poslovnih prostorov. Omogoča nam nadzor električnih naprav preko omrežij LAN/WAN



Slika 3.3: Senzor TEMPer1

(internet) [5]. Napravo priključimo v lokalno omrežje, v katerem želimo nadzirati naprave. To nam omogoča nadzor preko LAN omrežja, interneta ter mobilnega telefona ali tabličnega računalnika s sistemom Android. Nadzorna enota nudi programsko opremo za razvijalce. Do nadzorne enote DAEnetIP2 lahko dostopamo preko spletnega brskalnika in to z vpisom IP naslova nadzorne enote v naslovno vrstico. Primer modula lahko vidimo na sliki 3.4.



Slika 3.4: Denkovi DAEnetIP2 in 8 enotni rele modul

Poglavje 4

Razvoj sistema

V tem poglavju je podrobno predstavljena notranja zgradba sistema in komponente, ki ga sestavljajo. Opisana je implementacija nekaterih komponent. Pri tem so opisana tudi orodja in tehnologije uporabljene pri načrtovanju in implementaciji.

4.1 Orodja, protokoli in tehnologije uporabljene pri razvoju

V tem poglavju si bomo podrobno ogledali tehnologije in programska orodja, ki smo jih uporabili pri razvoju sistema. Ogledali si bomo tehnologije, ki so uporabljene pri izmenjavi podatkov. Pri tem se bomo osredotočili na spletne storitve in uporabljene protokole. Prav tako si bomo ogledali razvojna orodja, katera so bila uporabljena pri razvoju rešitve.

4.1.1 Tehnologije

4.1.1.1 Spletne storitve

Spletna storitev (ang. web service) je programska storitev v spletu, ki jo je mogoče poklicati s standardnimi spletnimi protokoli. Sestavljena je iz ene ali

več spletnih metod ali funkcij. Spletne storitve za komunikacijo z odjemalcem navadno uporabljajo protokol SOAP.

Spletne storitve so kot splošna tehnologija neodvisne od:

Programskega jezika Spletne storitve lahko pišemo v poljubnem programskem jeziku, ne glede na platformo operacijskega sistema.

Vrste aplikacije Narava aplikacije ne vpliva na možnost razvoja ali uporabe spletne storitve. Tako jih je mogoče uporabiti v namiznih aplikacijah, spletnih aplikacijah, na spletnih straneh ali pri drugih spletnih storitvah.

Operacijskega sistema Za vsak operacijski sistem obstajajo že izdelani objektni modeli, ki omogočajo pošiljanje sporočil SOAP in posledično uporabo tehnologije spletnih storitev. Tako lahko spletne storitve razvijamo in uporabljamo na vseh operacijskih sistemih podjetja Microsoft, prav tako pa tudi na vseh operacijskih sistemih UNIX in preostalih.

Strojne platforme Glede na neodvisnost tehnologije od operacijskih sistemov je mogoče sklepati, da je uporaba prav tako neodvisna od spodaj ležeče strojne platforme. Osnovna ideja spletnih storitev je, da jih je mogoče uporabiti in izdelati na poljubni strojni platformi, operacijskem sistemu, s poljubnim programskim jezikom. Možnost uporabe ni omejena na namizne ali spletne aplikacije. Odjemalec spletne storitve je tako lahko spletna stran, namizna aplikacija, aplikacija mobilne naprave ali druga spletna storitev.

Osnovna lastnost spletnih storitev je neodvisnost od uporabljene strojne in programske platforme. Spletne storitve predstavljajo sporočilno osnovan programski način komunikacije in so namenjene komunikaciji med dvema ali več programskimi sistemi. Pomembno je, da spletne storitve ne razumemo kot uporabniški mehanizem, ker niso namenjene komunikaciji med uporabnikom in računalnikom, pač pa medračunalniški komunikaciji.

SOAP SOAP je lahek protokol za izmenjavo informacij v decentraliziranem, porazdeljenem okolju. Protokol temelji na XML formatu in je sestavljen iz treh delov: ovojnice, ki definira ogrodje za opis, kaj je v sporočilu in kako ga obdelati; niza kodirnih pravil za izražanje primerov aplikacijsko definiranih podatkovnih tipov; konvencije za predstavitev oddaljenih proceduralnih klicev in odgovorov. Potencialno SOAP lahko uporabljamo tudi v kombinaciji z drugimi protokoli.

XSD Shema je abstraktna struktura za diagramsko upodobitev množice podatkov. Določa vrstni red oznak v dokumentu XML, označuje polja, ki so obvezna ali se lahko ponovijo, določa podatkovne tipe in tako naprej. Shema je pomembna, ker lahko z njo potrdimo, da je dokument veljaven glede elementov, atributov, strukture in tipov podatkov.

WSDL Za uspešno implementacijo spletnih storitev je potreben tudi neke vrste opis funkcionalnosti, ki odjemalcu poda zmožnosti spletne storitve. Trenutno je to podano v obliki standarda WSDL (Web Service Description Language), ki definira prisotne metode, njihove parametre in tipe parametrov, na katere se spletna storitev odziva. WSDL je zapisan v standardu XML, za sistem tipov pa uporablja standard "XML Schema".

4.1.1.2 Mobilni operacijski sistem Android

Android je odprtokodni operacijski sistem namenjen pametnim telefonom in tabličnim računalnikom. Njegova zgradba temelji na Linuxovem jedru. Namenjen je različnim uporabnikom ne glede na njihove izkušnje in znanja.

Za razvoj Androida je najzaslužnejše podjetje Google, ki je vzel pod okrilje podjetje Android. Prva različica je prišla na trg s prvim pametnim telefonom 9. februarja 2009 [6]. Bila je zelo okrnjena, kasneje pa so se začeli vrstiti popravki in nove verzije sistema.

Ker je sistem brezplačen, je proizvajalcem omogočil znižanje stroškov in posledično so uporabnikom na voljo cenejše in bolj funkcionalne mobilne naprave ter tablični računalniki. Naslednja pomembna lastnost sistema je, da omogoča razvijalcem več možnosti za razvoj programske opreme. An-

droid se lahko pohvali z uspešnim uporabniškim vmesnikom. Njegove glavne prednosti so hitra odzivnost, enostavnost in preglednost. Omogoča upravljanje preko zaslona in podpira večopravilnost, kar pomeni, da aplikacije lahko tečejo v ozadju sistema. Uporabnik le-te lahko hitro in enostavno nalaga, četudi nima na voljo osebnega računalnika.

4.1.1.3 SQLite

SQLite je odprtokodna podatkovna baza, ki podpira običajne lastnosti relacijskih podatkovnih baz, kot so npr. sintaksa SQL (angl. SQL syntax), transakcije itd., poleg tega pa med delovanjem porabi zelo malo delovnega pomnilnika (okoli 250 KB). V nasprotju z drugimi sistemi za upravljanje baz podatkov SQLite ni ločen proces, temveč sestavni del aplikacije kot programska knjižnica. Za razliko od ostalih relacijskih baz pa podpira nekoliko manj podatkovnih tipov za hranjenje podatkov. Med njimi so nizi znakov, cela števila, decimalna števila in poljubna binarna vrednost.

Slabost majhnih zahtev SQLite podatkovnih baz je tudi v tem, da ni nobenega preverjanja vnesenih podatkov. Torej lahko napišemo nek znakovni niz v polje tipa INTERGER in obratno, kar lahko privede do različnih napak med samim izvajanjem aplikacije. Pri implementaciji sistema smo uporabili knjižnico System.Data.SQLite, ki je specializirana verzija SQLite za okolje .NET.

4.1.2 Protokoli

4.1.2.1 SNMP

Protokol SNMP (Simple Network Management Protocol) je protokol za upravljanje zapletenih omrežij TCP/IP. Ta omogoča skrbnikom upravljanje in konfiguriranje naprav v omrežju z osrednjega računalnika brez dodatne programske opreme. SNMP je definiran na aplikativnem nivoju TCP/IP modela. Tipične naprave, ki podpirajo SNMP so omrežna stikala, usmerniki, tiskalniki, modemi in druge podobne naprave. V našem primeru tudi nadzorna

rele enota.

SNMP izpostavlja upravljanje podatkov v obliki spremenljivk, ki opisujejo konfiguracijo sistema. Protokol dovoljuje tudi aktivne upravljalvske naloge, kot so spreminjanje in uporaba novih nastavitev s pomočjo daljinske spremembe te spremenljivke. Spremenljivke dostopne prek SNMP so organizirane hierarhično. Ta hierarhija in drugi metapodatki (npr. vrsta in opis spremenljivke) so opisana zbirka informacij imenovana MIB (Management Information Base). Vsaki spremenljivki pripada OID (Object Identifier) in določa parameter na napravi, ki jo nadzorujemo. Sestavljen je iz označb organizacij za standardizacijo. Primer OID identifikatorja: 1.3.6.1.4.1.9.3.3.1. SNMP naprave, ki jih upravljamo, poznajo tri osnovne SNMP ukaze: Read, Write, Trap.

Napravo, ki jo upravljamo s protokolom SNMP vsebuje tri ključne komponente:

Nadzorovana naprava Naprava ima implementiran SNMP vmesnik, ki omogoča enosmerno (samo branje) ali dvosmerno (branje in spreminjanje) komunikacijo. Vsebuje t.i. SNMP agenta.

SNMP agent Agent je program, ki deluje na nadzorovani napravi. Podatke, ki jih naprava shranjuje, agent pretvori v SNMP razumljivo obliko.

NMS (Network Management System) Skupek naprav in programske opreme, ki nadzirajo in kontrolirajo naprave. NMS procesira in shranjuje vse potrebne informacije. V enem omrežju lahko istočasno obstaja več NMS-jev.

4.1.3 Razvojna orodja

V tem poglavju si bomo podrobno ogledali konkretna programska orodja, ki smo jih uporabili pri razvoju rešitve. Uporabili smo odprtokodna ali pa brezplačna orodja za nekomercialno uporabo. V nadaljevanju si bomo pogloblje ogledali vsakega izmed njih.

4.1.3.1 Visual studio 2010

Microsoft Visual Studio je integrirano razvojno microsoftovo okolje (IDE). Razvojno okolje je namenjeno razvoju programov, ki tečejo pod okoljem Windows. Uporabljamo ga lahko za razvoj upravljalnih aplikacij, grafičnih vmesnikov, spletnih strani, spletnih storitev in mobilnih aplikacij [7]. Visual studio omogoča programiranje v naslednjih programskih jezikih C/C++, Visual Basic .NET, C#, F#, Python, Ruby. Prav tako podpira XML / XSLT, HTML / XHTML, JavaScript in CSS.

Glavne značilnosti razvojnega okolja:

Urejevalnik kode Napreden urejevalnik kode z vgrajeno tehnologijo IntelliSense, ki nam med programiranjem prikaže razrede in metode, ki jih imamo na voljo. Preurejevalnik kode (angl. Refactoring) omogoča izboljšavo berljivosti in strukturo kode.

Izdelava grafičnih vmesnikov Enostavno izdelovanje zaslonских vmesnikov po principu WYSIWYG.

Razhroščevalnik Ena izmed najbolj pomembnih lastnosti Visual Studia je možnost izvajanja sprehajanja po aplikaciji vrstico za vrstico. Tako hitreje odpravimo napake v kodi.

Organiziranost Organizacija kode v projekte in projektov v rešitve.

Razširljivnost Razvijalci lahko razvijajo dodatke za Visual studio, ki razširijo njegove zmožnosti. Lahko jih dodamo v obliki makrov, vtičnikov ali paketov. Razširitve vključujejo podporo za nove programske jezike ali podporo za sisteme za nadzor izvirne kode (Subversion in Team Foundation Server).

4.1.3.2 Eclipse in ADT vtičnik

Eclipse je večjezikovno orodje za razvoj programske opreme in je trenutno najbolj razširjeno za razvoj Android aplikacij. Gre za odprtokodno razvojno

okolje, ki omogoča programiranje v številnih programskih jezikih. Ker ima dobro razvit sistem za vtičnike, se jih je sčasoma nabralo kar veliko število. Ti podpirajo razvoj opreme v različnih programskih jezikih, kot so Java, Ada, C, C++, COBOL, Haskell, Perl, PHP, Python, R, Ruby, Clojure. Izbran je bil na podlagi široke podpore za razvoj Android aplikacij in odlične kompatibilnosti z razvojnimi orodji Android. Podjetje Google tudi uradno podpira razvojno okolje Eclipse.

ADT vtičnik je razširitev Java okolja, tako da podpira pisanje aplikacij za android. Vtičnik je razvil Google in ga tudi redno vzdržuje. Omogoča pisanje kode, urejanje resource datotek in kreiranje grafičnega vmesnika vizualno. Prav tako ima odlično podporo za razhroščevanje tako na fizični napravi kot Android emulatorju.

4.1.3.3 Wireshark

Wireshark je odprtokodna programska oprema za paketno analizo. Uporablja se za odpravljanje težav z omrežjem, za analize, izobraževanje in razvoj programske opreme in komunikacijskih protokolov [8]. Wireshark omogoča uporabniku, da na izbranem omrežnem vmesniku zajame ves promet, tudi tak, ki je namenjen drugim naslovom. Aplikacija ima bogat uporabniški vmesnik, ki olajša vizualizacijo zajetih paketov. Prav tako omogoča napredne možnosti filtriranja in shranjevanja zajetih paketov v datoteke.

Wireshark je programska oprema, ki "razume" strukturo različnih protokolov povezovanja. Sposoben je prikazati enkapsulacijo in področja, skupaj z njihovimi pomeni različnih paketov, ki jih imajo različni protokoli omrežja. Analiza nekaterih razširjenih protokolov je enostavna, saj ima program specializirana orodja za posamezne protokole, kot so http, VoIP. Kot rečeno, pa Wireshark omogoča implementacijo lastnih vtičnikov za analizo lastnih protokolov.

4.2 Strežnik

V tem poglavju je predstavljena strežnika aplikacija, ki zajema senzorske podatke, nadzira strojno opremo in izvaja odločitveno logiko. Strežniška aplikacija je implementirana kot običajen program s konzolnim uporabniškim vmesnikom. Program vsako pomembnejšo komponento izvaja v svoji niti. Sočasno lahko zajema podatke kot tudi obdeluje odjemalčeve poizvedbe po trenutnem stanju ali poizvedbe za senzorske vrednosti v preteklem časovnem obdobju.

4.2.1 Opis zgradbe aplikacije

Strežnik je aplikacija napisana v programskem jeziku **C#** za okolje Windows. Aplikacija je implementirana kot običajen program s konzolnim uporabniškim vmesnikom. Dodana pa je tudi možnost, da aplikacija teče v načinu Windows storitve (Windows Service). V tem načinu delovanja ni potrebna interakcija z uporabnikom in je tako primerna za strežnike, ker ni potrebno, da so uporabniki stalno prijavljeni v sistem.

Strežnik sestavlja več med sabo povezanih komponent. Komponenta sistema je lahko implementirana kot programski modul ali kot samostojna komponenta, ki se časovno neodvisno izvaja od drugih komponent. Vsaka komponenta, ki je časovno neodvisna (npr. zajem vrednosti in odločitvena logika) se izvaja v svoji niti.

Shranjevanje in branje nastavitev Aplikacija je odvisna od lokalne strojne opreme, zato smo podatke o senzorjih in relejih zapisali v nastavitveno datoteko. Nastavitvena datoteka je v XML formatu in jo lahko urejamo s katerim koli programom, ki omogoča urejanje tekstovnih datotek. V nastavitvah so definirane skupine senzorjev, naslov in tip senzorja iz katerega beremo podatke. Poleg tega so vključeni tudi podatki o nazivu in barvi senzorja. Barva senzorja se kasneje uporabi za izris grafov na odjemalčevi strani. Podobno velja tudi za rele enote. Poleg strojne opreme so v nastavitvah de-

finirani tudi vsi sprožilci za odločitveno logiko.

Zajem vrednosti senzorjev Komponenta poskrbi, da se iz vsake naprave, ki je bila določena v nastavitvah, zajame vrednosti glede na vrsto naprave. Prebrane vrednosti je potrebno pretvori v °C . Pretvorba pa je odvisna od strojnega tipa senzorja. Komponenta tudi skrbi za napačno prebrane vrednost, ki so izven območja senzorja. Prav tako se beleži tudi morebitna odsotnost ali okvara senzorja.

Nadzor rele enot Komponenta skrbi za komunikacijo z rele napravami in ima podobno abstrakcijo strojne opreme kot temperaturni senzorji. Komunikacija je tukaj dvosmerna, saj pošiljamo ukaze za vklop in izklop releja.

Odločitvena logika Komponenta periodično preverja ali so definirani pogoji v sprožilcih izpolnjeni. Pogoji se lahko preverjajo glede na vrednost točno določenega senzorja ali pa glede na trenutni čas. Ko so vsi pogoji izpolnjeni, se izvede procedura, ki je definirana v sprožilcu. Ta procedura je največkrat vklop ali izklop določenega releja.

Podatkovni vmesnik Skrbi za shranjevanje zajetih vrednosti v podatkovno bazo zaradi vpogleda na zahtevo tudi za preteklo obdobje. Shranjujejo se tudi informacije o sproženih dogodkih preko odločitvene logike in ostale napake. Komponenta za dostop do podatkovne baze uporablja zunanjo knjižnico "System.Data.SQLite". Podatkovna baza se nahaja v datoteki s končnico ".db3".

Oddaljen nadzor Komponenta skrbi za strežbo zahtev odjemalcev in je neodvisna od drugih komponent v strežniški aplikaciji. Omogoča spreminjanje delovanja strežnika, urejanje sprožilcev in povpraševanje o stanju senzorjev in relejev. Ukazi in funkcije so na voljo kot spletna storitev. Spletna storitev je implementirana kot ugnезden HTTP strežnik znotraj aplikacije.

4.2.2 Gonilniki za nadzor strojne opreme

Sistem komunicira z zelo različno strojno opremo, za katero ni enotnega standarda ali protokola. Večina uporabljene strojne opreme uporablja svoj lasten protokol, brez javne dokumentacije. Taka strojna oprema deluje samo s

priloženo programsko opremo. Komunikacija strojne opreme z računalnikom je sicer standardna (TCP/IP, USB, serijski vmesnik), kljub temu pa je bilo potrebno vložiti veliko dela v analizo protokola, ki ga uporablja posamezen proizvajalec. Pri temperaturnih senzorjih je bilo potrebno ločiti analizo na komunikacijo z nadzorno napravo na katero se povezujemo in prebrane vrednosti posameznih senzorjev.

Zaradi velike raznolikosti strojne opreme smo se odločili za čim bolj abstraktno strukturo in implementacijo. Ko program želi prebrati podatek iz senzorja, pokliče funkcijo vmesnika, ta pa glede na implementacijo komunicira s strojno opremo. Sistema ne zanima način ,kako se pridobi podatek. Abstrakcija nam omogoča, da temperaturne vrednosti lahko beremo iz strojne opreme, datoteke ali celo spletne strani. Kot dokaz koncepta smo implementirali senzor, ki trenutno temperaturo prebere iz spletne strani Agencije RS za okolje (ARSO). Implementacija komunikacije z rele moduli je podobna, le da se tam uporablja branje in zapis podatkov.

Implementirani gonilniki za nadzorne naprave senzorjev:

- 1W340A (TCP/IP),
- TEMPer1 (USB),
- ARSO meteorološke postaje (zajem vrednosti preko interneta),
- branje iz datoteke.

Implementirani gonilniki za temperaturne senzorje:

- DS18B20,
- FM75,
- TEMPer1Sensor.

Implementirani gonilniki za nadzor rele modulov:

- Denkovi DAEnetIP2 (TCP/IP, SNMP),
- Denkovi USBFTDI (USB).

4.2.2.1 Zajem temperaturnih vrednosti

Če želimo prebrati vrednosti senzorjev, je to potrebno narediti preko nadzorne naprave. Nadzorna naprava ima priključenih več senzorjev, komunikacijo z njimi pa sistem opravlja periodično in samodejno. Po vzpostavljeni povezavi z napravo, pošljemo zaporedje ukazov, s katerimi sporočimo napravi, da želimo prebrati vrednosti senzorjev. Kot odgovor dobimo tabelo vrednosti za vse priključene senzorje. Prejete podatke je potrebno razčleniti in ustrezno pretvoriti. V primeru da komuniciramo z USB senzorjem, je postopek podoben, le da kot odgovor dobimo vrednosti samo za en senzor.

V nadaljevanju se bomo osredotočili na primer implementacije branja podatkov iz nadzorne enote 1W340A in temperaturnega senzorja DS18B20. Do nadzorne naprave dostopamo preko IP naslova in vrat 5200. Vzpostavimo TCP povezavo, ki jo držimo odprto toliko časa, dokler ne preberemo zelenih vrednosti. Ko je povezava uspešno vzpostavljena, pošljemo zaporedje ukazov, ki napravi povedo, da želimo prebrati vrednosti senzorjev. Zaporedje ukazov smo pridobili z analizo paketov, ki se prenašajo z uporabo programske opreme priložene napravi. Pri tem smo uporabili program Wireshark, kateri nam je zelo olajšal analizo protokola. Konkretno zaporedje ukazov za nadzorno napravo 1W340A in senzorje DS18B20 je sestavljeno iz treh zlogov bajtov 0xBB, 0x80, 0x00. Za druge naprave in senzorje se ukaz razlikuje vsaj v enem zlogu. Kot odgovor dobimo polje bajtov. Tukaj je bila potreba podrobna analiza pomena podatkov in kako so med sabo povezani. Na sliki 4.1 lahko vidimo primer prejetih podatkov in njihovo logično predstavitev. Kot lahko opazimo, dobimo v podatkih število priključenih senzorjev in dva zloga temperaturne vrednosti za vsak senzor, ki si sledijo en za drugim.

Posamezno vrednost senzorja je potrebno pretvoriti v $^{\circ}\text{C}$. Pri pretvorbi smo si pomagali z dokumentacijo senzorja [3]. Upoštevati je potrebno ločljivost senzorja in predznak vrednosti. Iz prejetih zlogov odstranimo nepotrebne bite in jih ustrezno zamaknemo. Rezultat pomnožimo z ločljivostjo senzorja. Dodatno še prištejemo kalibracijsko vrednost in preverimo ali je prebrana vrednost v območju, ki ga senzor podpira. Primer kode, ki pretvori prebrano

0x00	
0x02	Število priključenih senzorjev
0x121	
0x04	Senzor #1
0x94	
0x02	Senzor #2
...	
...	

Slika 4.1: Primer prejetih podatkov nadzorne enote 1W340A

vrednost v °C, lahko vidimo na sliki 4.2.

```

/// <summary>
/// Specifična metoda za branje temperature iz prejetega medpomnilnika.
/// </summary>
public void ReadFromBuffer(ref byte[] buffer)
{
    int index = 4 + HardwareIndex * 2;
    switch (HardwareType)
    {
        case "DS18B20":
        {
            if (buffer[index] <= 7)
                Value = (buffer[index] * 256) + buffer[index + 1];
            else
                Value = -1 * (65536 - ((buffer[index] * 256) + buffer[index + 1]));
            Value *= 0.0625;
            Value += Correction;
        }
        break;
    }
    Status = SensorStatus.Online;

    if (Value > 120 || Value < -40)
        Status = SensorStatus.Offline;

    ValueTime = DateTime.Now;
}

```

Slika 4.2: Programska koda funkcije za pretvorbo vrednosti

4.2.2.2 Nadzor rele modulov

Rele enote nadziramo preko rele nadzorne enote. Nadzorna enota ima priključenih več rele enot. Število rele enot, ki jih lahko nadzorujemo, se giblje med 4 in 24, odvisno od modela [5].

V nadaljevanju se bomo osredotočili na primer implementacije nadzora releja iz nadzorne enote DAEnetIP2 s pomočjo protokola SNMP. Uporabili smo knjižnico SharpSnmpLib, ki je napisana za uporabo v NET programskem okolju in nam zelo olajša komunikacijo z nadzorno napravo. Potrebno je poznati IP naslov nadzorne naprave in OIB. OIB smo pridobili iz specifikacij naprave, ki jih je priložil proizvajalec. V knjižnici se nahajajo funkcije, s katerimi beremo vrednost iz posamezne enote. Prebrana vrednost je 32 bitno število, ki si ga moramo predstavljati kot opis stanja celotnega rele modula. Vsak bit določa stanje rele enote. S pomočjo AND in OR mask sestavimo 32 bitno vrednost v obliki celoštevilске vrednosti. To vrednost nato pošljemo napravi.

4.2.3 Shranjevanje podatkov

Pomnjenje zajetih podatkov je pomembni del sistema. Sam mora omogočati možnost pregleda zajetih vrednosti v preteklem časovnem obdobju. Po daljšem času delovanja se količina podatkov zelo poveča. Zaradi večje količine podatkov pa pri poizvedbah za daljše časovno obdobje v preteklosti, lahko hitro nastane problem. Eden izmed pogojev pri izbiri shranjevanja podatkov je, da mora biti dostopen čas neodvisen od količine podatkov. To zahtevo lahko zadovoljimo samo, če imamo podatke indeksirane. Odločili smo se za SQLite. SQLite ima večino funkcionalnosti, ki jih ostali večji sistemi za upravljanje s podatkovnimi bazami (SUBP) nimajo, kljub temu pa potrebuje za svoje delovanje zelo malo sistemskih sredstev.

Kot vmesnik z SQLite bazo, smo uporabili knjižnico System.Data.SQLite. To je specializirana implementacija vmesnika za NET okolje. Vmesnik za bazo pa je enak kot pri drugih SUBP. Za enostaven zapis vrednosti v tabelo je

najprej potrebno odpreti povezavo z bazo, ki je v našem primeru datoteka. V konfiguracijskem nizu povemo, kje se nahaja ta datoteka, zraven pa vključimo še različico. Nato inicializiramo ukaz z SQL poizvedbo, v našem primeru INSERT stavek. Določimo parametre in SQL ukaz izvedemo.

4.2.4 Implementacija odločitvene logike

Ključni del sistema je samodejno odločanje in izvajanje procedur glede na zunanje pogoje, zato smo velik del načrtovanja posvetili izdelavi te komponente. Komponenta deluje v povezavi z zajemanjem temperaturne vrednosti v fiksnem časovnem intervalu. Ob sistemskem dogodku časovnika se pridobijo vrednosti senzorjev stanja rele modulov. Če je sistem v avtomatskem načinu delovanja, se preverijo pogoji znotraj definiranih sprožilcev. V primeru, da ima kateri od sprožilcev izpolnjenje vse pogoje, se izvršijo procedure definirane na sprožilcu. Običajno je to vklop ali izklop rele enote.

Sprožilec je element, ki vsebuje enega ali več pogojev. Pogoji so lahko primerjalni glede na vrednost senzorja, releja ali časovni. Šele, ko so vsi pogoji izpolnjeni, se izvrši ena ali več procedur. Ukazi, ki se izvajajo, so lahko vklop ali izklop rele modula (najpogostejše) ali zagon nekega tretjega programa.

Pogoj je logičen izraz, ki je v nekem trenutku lahko izpolnjen ali ne. Torej imamo opravka z Boolovo algebro. Sistem trenutno podpira dve vrsti pogojev:

Vrednostni Pogoj primerja vrednost točno določenega senzorja ali releja s primerjalno vrednostno. Primer takega pogoja: vrednost senzorja mora biti večja od 30 stopinj. Če senzorju pripišemo unikatno oznako S1, potem je izraz sledeč $S1 > 30$. Za rele pa se lahko vprašamo ali je vklopljen ali ne.

Časovni Pogoj primerja trenutni čas z vpisanim časom intervalom. Primer takega pogoja: želimo, da je sprožilec aktiven od 13.00 do 16.00.

Ukaz je procedura, ki se izvrši, ko so vsi pogoji sprožilca izpolnjeni. Sistem trenutno podpira tri vrste ukazov. Med njimi je najpomembnejši ukaz vklop ali izklop rele modula preko vmesnika za nadzor rele modulov.

Dogodek releja Vklop ali izklop releja. Dogodek uporablja vmesnik za nadzor rele modulov opisan v poglavju 4.2.2.2.

Dogodek zagona programa Zagon katerega koli drugega programa na strežniku s fiksnimi parametri. To je predvsem uporabno, če želimo povezati obstoječe sisteme ali obveščanje uporabnika preko e-pošte.

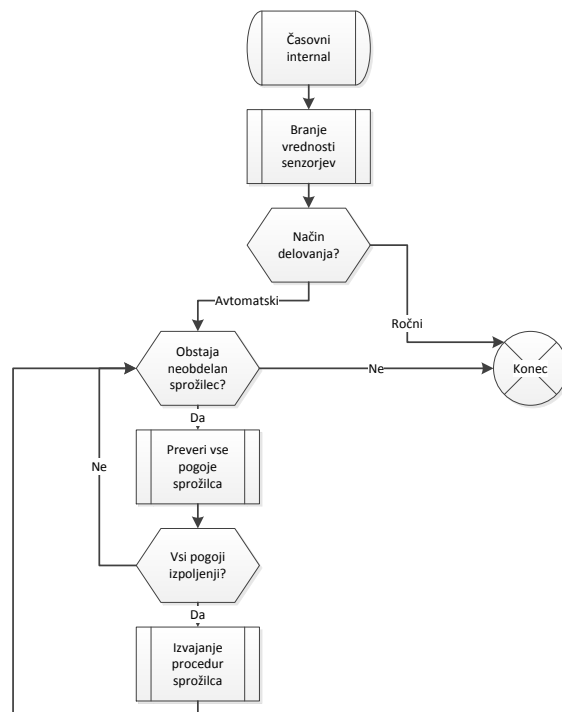
Diagram poteka odločitvene logike lahko vidimo na sliki 4.3. Za lažjo predstavo pa definirajmo še primer konkretnega sprožilca. Imamo senzorje S1 in S2, in releja R1 in R2. Želimo vklop rele enote R1 v času med 6.00 in 12.00 uro in ko vrednost senzorja S1 pada pod 20°C. Potrebno je definirati sprožilec, ki ima dva pogoja in eno proceduro za vklop rele enote.

Če so vsi pogoji na sprožilcu izpolnjeni, zaženemo procedure za vklop releja. Velja opozoriti, da kadar ima več sprožilcev istočasno uporabljeno rele enoto R1, se med vsemi sprožilci dela logični OR. To v praksi pomeni, da je potreben najmanj en primeren sprožilec in rele bo ostal vklopljen. Torej delamo logični OR med sprožilci in logični AND med pogoji znotraj sprožilca.

4.2.5 Oddaljeni nadzor

Sistem omogoča nadziranje preko spletnih storitev. Spletne storitve so bile izbrane zaradi splošne razširjenosti na vseh platformah in uporabe standardiziranih protokolov. Uporabili smo dva različna modela spletnih storitev. Za natančno definirane funkcije nadzora smo uporabili SOAP protokol. SOAP natančno definira funkcije, strukturo podatkov in XML pakete preko katerih poteka komunikacija. Za prikazovanje združenih podatkov v obliki grafa pa smo uporabili REST model.

- Stanje, nadzor (vrednosti senzorjev, zgodovina podatkov, nadzor rele naprav),



Slika 4.3: Diagram poteka za odločitveno logiko.

- Združeni podatki v obliki grafa. Rezultat je bitna slika vrednosti senzorjev v odvisnosti od časa.

Obe storitvi sta realizirani kot procesa znotraj aplikacije. Tak način implementacije omogoča WCF (Windows Communication Foundation), podsystem .NET framework-a. Omogoča različne vrste komunikacije, kot so nizkonivojski TCP/IP, spletne storitve, HTTP, RPC (Remote procedure call), itd. Strukture in funkcije programskega razreda izpostavimo kot vmesnik (interface). Sistem sam poskrbi za ustrezno nastavitve storitve, pretvorbo poslanih in prejetih podatkov na ustrezno obliko.

Storitev za pridobivanje stanja in nadzor ima deset funkcij. Vsaka od njih tudi vrača podatke. Ti so lahko sezname vrednosti ali pa samo stanje, da je bila operacija uspešna. Storitev uporablja vrata 8000. Klicanje funkcij na spletnih storitvah poteka preko HTTP protokola in POST ukaza.

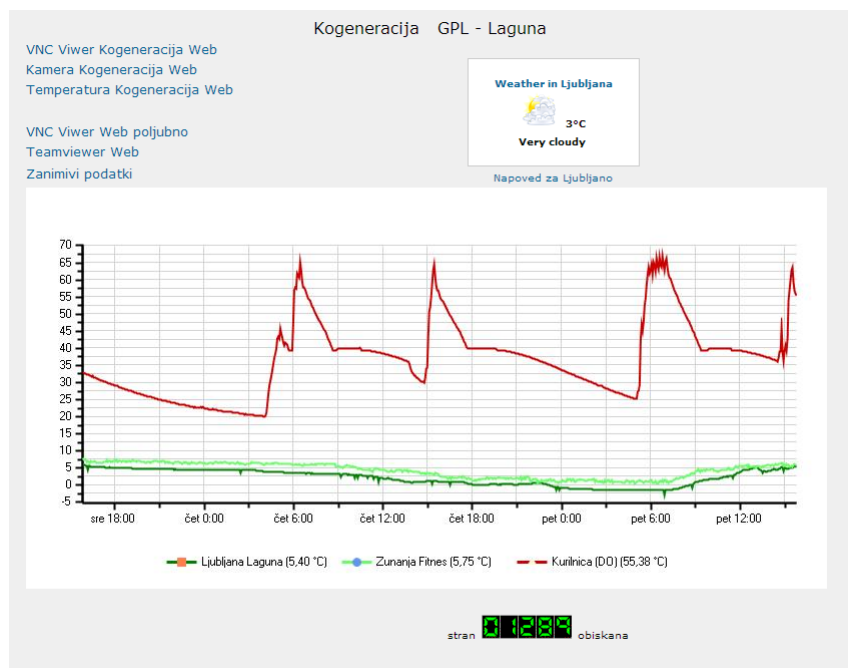
Druga storitev pa ima samo eno funkcijo. Uporablja generičen HTTP protokol in GET. Parametri, ki jih pošljemo sistemu, se nahajajo v naslovu, ki ga zahtevamo preko HTTP GET zahteve. Ker je rezultat storitve slika, jo lahko vključimo na katero koli spletno stran. Primer uporabe take slike lahko vidimo na sliki 4.4.

4.2.5.1 Zagotavljanje varnosti

Oddaljen dostop do podatkov in nadzor predstavlja tudi varnostne probleme. Nepooblaščen oseba lahko pridobi aplikacijo za nadzor ali celo odtuji mobilno napravo. Vsako poseganje nepooblaščen osebe ima lahko negativne posledice tako za opremo kot tudi na stroškovni strani.

V primeru, da se želimo zavarovati pred takimi posegi oziroma jih preprečiti, je potrebno:

- onemogoči uporabnika, ki je odtujil tako napravo;
- onemogočiti uporabniki, ki se je seznanil s protokolom in lahko vseeno pošilja zahteve preko WS.



Slika 4.4: Vpeti podatke na spletni strani pridobljeni preko spletne storitve.

Ker je sistem dostopen preko interneta in ker različne tehnologije (Windows, Android) ne podpirajo istih načinov avtorizacije, smo uvedli enostaven način preverjanja preko zakodiranega gesla poljubne dolžine, ki rešita oba opisana problema. Zahtevano geslo je pogoj za vstop v aplikacijo. Vnos le-tega je potreben tako na namizni aplikaciji kot tudi na mobilni. Vpisano geslo se nato transformira s pomočjo zgoščevalne funkcije MD5. Izvleček gesla pa se pošlje kot dodaten parameter pri vsakem klicu spletne storitve sistema. Isti izvleček gesla je zapisan tudi na strežniški strani, v nastavitveni datoteki. Še vedno obstaja možnost zlorabe sistema, vendar tak sistem te možnosti zelo omeji. Človeški faktor pa je še vedno ključen.

4.3 Odjemalci

V tem poglavju so predstavljeni različni odjemalci podatkov, ki jih ponuja sistem kot storitev. Opisana je implementacija aplikacij za različne operacijske sisteme. Pri vsaki aplikaciji je opisana tudi funkcionalnost in uporaba funkcij.

4.3.1 Opis splošnih funkcionalnosti

Implementirali smo dve različni aplikaciji, prva teče na okolju windows in je namenjena napredni spremljavi podatkov, druga pa v mobilnem operacijskem sistemu Android. Funkcionalnosti se malo razlikujeta, s tem da je mobilna omejena s funkcijami. V obeh aplikacijah pa lahko pregledujemo trenutno vrednost senzorjev, trenutno stanje rele modulov in splošno stanje sistema.

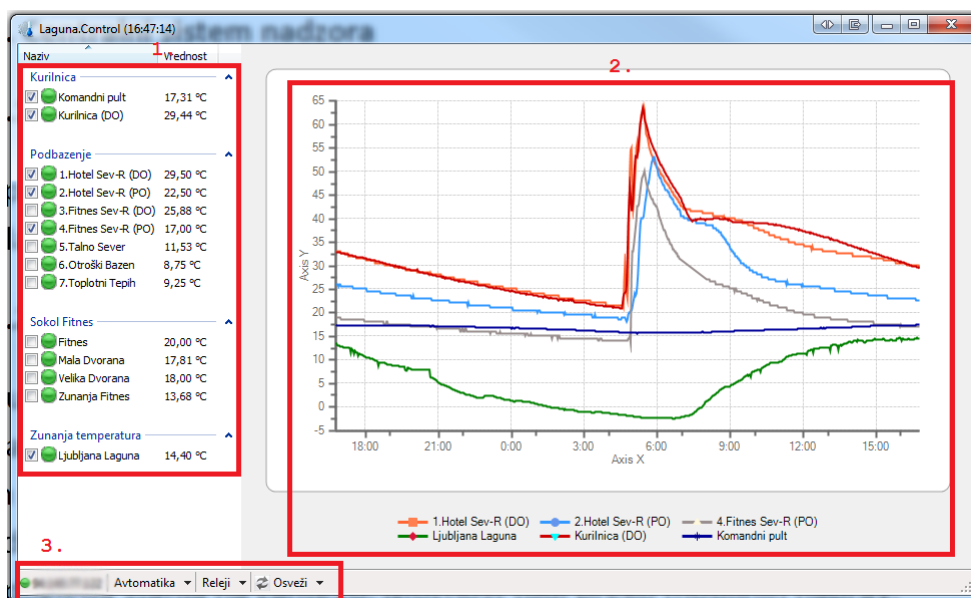
Obe aplikaciji omogočata tudi vpogled v zgodovino vrednosti senzorjev in vizualno predstavitev teh vrednosti v obliki grafa. Pri izpisu grafa lahko vedno izberemo, katere vrednosti senzorjev bodo prikazane. Naziv in barva posameznega senzorja je zaradi enotnosti definirana na strežniški strani.

4.3.2 Windows aplikacija

Aplikacija je napisana v programskem jeziku **C#** in deluje samo na operacijskem sistemu Windows. Uporabniški vmesnik je grafičen, kar uporabniku omogoča učinkovito upravljanje. Prav tako pa so bistvene informacije glede sistema vedno na voljo.

Zaslonsko sliko glavnega okna vidimo na sliki 4.5. Vmesnik je razdeljen na tri večje komponente.

1. Seznam priključenih senzorjev, njihovo stanje naziv in trenutna vrednost. Pred vsakim senzorjem imamo izbirno polje, s katerim lahko vklopimo ali izklopimo prikaz na grafu.
2. Vizualna predstavitev zgodovine vrednosti za izbrane senzorje. Obdobje za katerega so prikazane vrednosti lahko nastavimo preko predla-



Slika 4.5: Glavna zaslonska slika

ganih vrednosti kot so 1 ura, 3 ure, 12 ur, 1 dan ... itd. Imamo pa tudi možnost, da ročno definiramo obdobje. Obdobje lahko nastavimo kot datum in čas začetka in datum in čas konca.

- Orodna vrstica v kateri je vidno stanje povezave s strežnikom. Iz tega menija je možna sprememba načina delovanja, urejanja sprožilcev in nadzor rele enot, vendar mora biti strežnik v pravilnem načinu delovanja. Prav tako lahko vidimo, v kakšnem stanju je posamezna rele enota. Če je vklopljena, vidimo zraven naziva kljukico. Graf omogoča tudi samodejno osveževanje vrednosti na seznamu senzorjev in grafu.

4.3.2.1 Prilagoditve vmesnika

Windows aplikacija ima dva različna načina uporabe. V enem ima uporabnik (administrator) polni nadzor in lahko spreminja stanje sistema. V drugem pa samo pregled nad vrednostmi senzorjev in vpogled v zgodovino teh vrednosti. Način uporabe je določen preko nastavitve aplikacije.

Administratorski način je dovoljen samo na delovni enoti, ki se nahaja v komandi sobi, kjer imajo dostop samo pooblašene osebe. Navaden način, kjer se samo pregleduje vrednosti, je mišljen kot prikaz temperature v različnih prostorih v realnem času. Nekateri zaposleni pa imajo aplikacijo nameščeno tudi na lastnih računalnikih. Tam jo uporabljajo za spremljanje temperature v sistemu oziroma ugotavlja izrednih dogodkov.

4.3.2.2 Opis funkcionalnosti

Pregled zgodovine vrednosti

Če želimo vpogled v zgodovino vrednosti senzorjev v točno določenem obdobju, najprej izberemo senzorje, za katere želimo, da bodo prikazani. Nato v meniju izberemo "Obdobje". V oknu, ki se nam prikaže, pa vnesemo obdobje, za katero želimo pregledovati zgodovino. Glede na izbrano obdobje je potrebno počakati, da se podatki prenesejo.

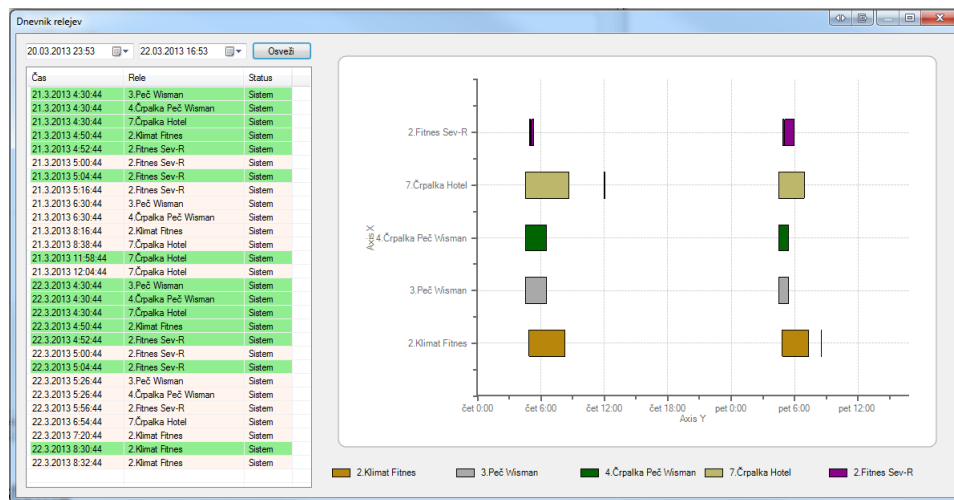
Pregled zgodovine stanj relejev

Če želimo vpogled v zgodovino stanj rele enot, v meniju izberemo "Dnevnik relejev". V okno, ki se nam prikaže, vnesemo obdobje, za katero želimo pregledovati zgodovino. Ob vstopu, nam kot prvo izbiro vedno predlaga obdobje prejšnjega dne. Kot rezultat dobimo seznam dogodkov vklopov in izklopov relejev. Med njimi so tudi podrobnosti časa vklopa in način, kako je bilo stanje rele enote spremenjeno. Stanje lahko spremeni bodisi sistem sam ali pa ga ročno spremeni uporabnik.

Na zaslonski sliki 4.6 vidimo tudi Ganttov diagram rele dogodkov. Na grafu lahko natančno vidimo, kdaj in koliko časa je bila posamezna rele enota vklopljena.

Pregled zgodovine proženja sprožilcev

Podobno kot pregled dogodkov releja imamo možnost pregleda pogojev sprožilcev. Čim pride do kakšne spremembe v pogojih sprožilca, se to zapiše v podat-



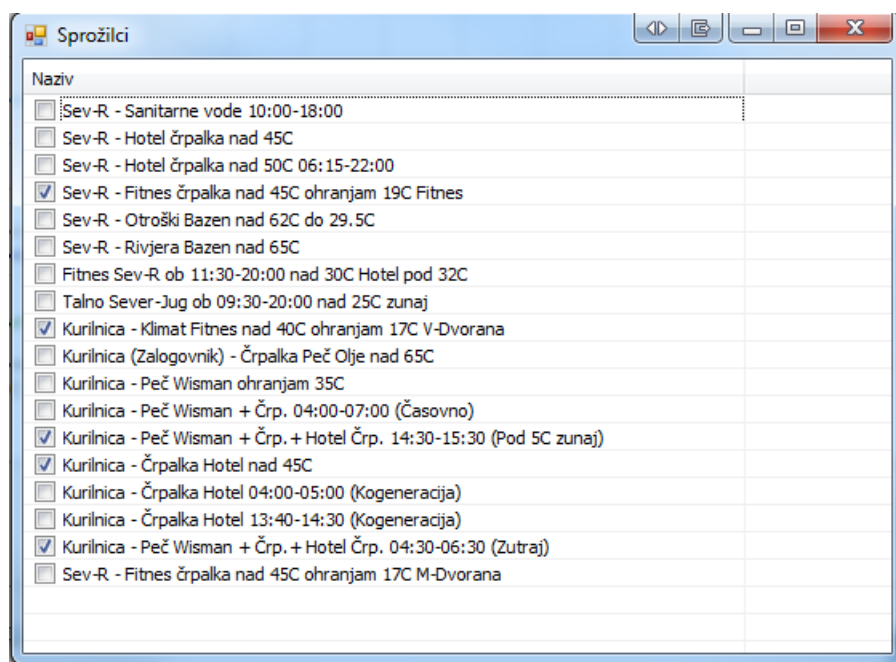
Slika 4.6: Ganttov diagram dogodkov rele enot

kovno bazo v sistemu. S tem pregledom lahko vidimo, kateri pogoji so bili izpolnjeni in kateri ne. To nam koristi v primerih, ko želimo ugotoviti, zakaj se je določen sprožilce izklopil.

Urejanje sprožilcev

Aplikacija ima možnost urejanja sprožilcev. Na seznamu vseh sprožilcev 4.7 vidimo njihove podatke in aktivnost. Aktivnost sprožilca pomeni ali se upošteva pri odločitveni logiki ali ne. Imamo lahko definiranih več podobnih sprožilcev, ki pa se uporabijo v različnih obdobjih. Vsakršna sprememba je takoj vidna na strežniku in hkrati shranjena tudi v nastavitveno datoteko na strežniku.

Na sliki vnosne zaslonske maske za urejanje sprožilca 4.8 vnesemo naziv, aktivnost in dodamo poljubno število pogojev. Za vsak pogoj se nam odpre drugo okno, kjer vnesemo relevantne podatke; npr za časovni pogoj, čas itd. Pri senzorskih pogojih nam ponudi seznam vseh priključenih senzorjev. Podobno storimo tudi za procedure. Pri senzorskih procedurah nam ponudi seznam vseh priključenih relejev.



Slika 4.7: Seznam sprožilcev in njihova aktivnost

4.3.3 Mobilna aplikacija Android

Napisana je za operacijski sistem Android v jeziku Java. Zagon aplikacije je možen na tabličnih računalnikih in mobilnih telefonih z operacijskim sistemom Android. Aplikacija omogoča spreminjanje trenutnih vrednosti senzorjev, stanja sistema in rele enot. Prav tako omogoča pregledovanje po zgodovini vrednosti in vizualizacijo podatkov na grafu. Grafični vmesnik se za različne mobilne naprave (mobitel, tablični računalnik) ne razlikuje. Gradniki na vmesniku se le ustrezno povečajo ali razširijo.

Za graf smo uporabili odprtokodno kontrolo "MapView", ki omogoča enostavno vizualizacijo podatkov [9]. Kontrola omogoča prepoznavanje gest na večdotičnem zaslonu. Uporabili smo gesto stisni (ang. "pinch to zoom") in premik za interakcijo z grafom.

Urejanje sprožilca

Naziv: Sev-R - Fitnes črpalka nad 45C ohranjam 19C Fitnes

☒ Omogočen ☐ Začasen

Dodaj Uredi Odstrani

Vrsta pogoja	Pogoj
Senzor	1.Hotel Sev-R (DO) [IP1] > 45,00 °C
Senzor	Fitnes [SF3] < 19,00 °C

Dodaj Uredi Odstrani

Rele dogodek	Status
2.Fitnes Sev-R [RELE2]	On

Shrani Prekliči

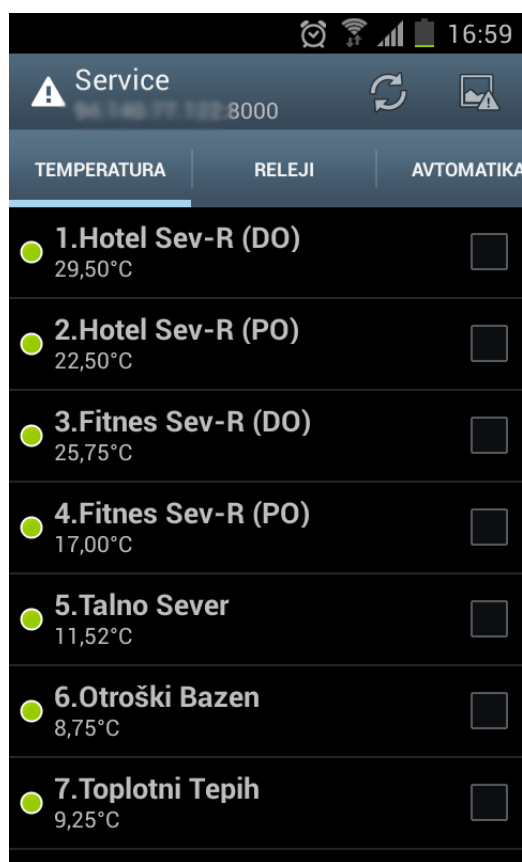
Slika 4.8: Vnosna maska za urejanje sprožilca

4.3.3.1 Splošno o mobilnih aplikacijah

Prenosne naprave kot so mobilni telefoni in tablični računalniku, so čedalje bolj dostopne in jih uporablja vedno več ljudi. Naprava dandanes lahko opravi isto delo, kot kateri koli drugi računalnik, s tem da je taka naprava vedno z nami. To je bil razlog, zaradi katerega smo se lotili razvoja aplikacije za mobilne naprave. Naprave omogočajo tudi interakcijo preko dotikov, kar pomeni, da se nam odpre veliko novih možnosti za interakcijo s podatki.

4.3.3.2 Opis funkcionalnosti

Ob vstopu v aplikacijo se nam prikaže seznam priključenih temperaturnih senzorjev, njihovo trenutno stanje in temperatura. Zaslonsko sliko lahko vidimo na sliki 4.9.

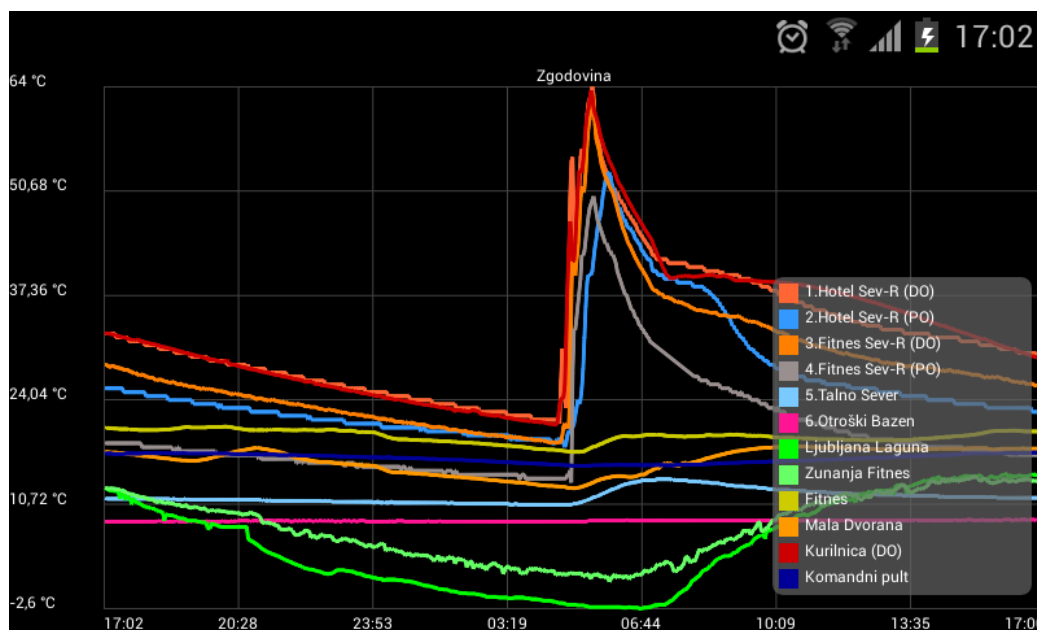


Slika 4.9: Seznam temperaturnih senzorjev

Aplikacija omogoča samodejno ali ročno osveževanje vrednosti. Avtomatsko osveževanje ne deluje, če je aplikacija zaprta ali minimizirana; s tem prihranimo baterijo, istočasno pa tudi količino prenesenih podatkov.

Pregled zgodovine vrednosti

Če želimo pregledovati zgodovino vrednosti za izbrane senzorje, jih izberemo tako, da nanje pritisnemo. Ko je senzor izbran, se na desni strani senzorja označi izbirno polje. Po končani izbiri v meniju izberemo "pregled zgodovine". Meni nam ponudi izbor obdobja, za katera želimo pridobiti vrednosti. Izbiramo lahko med predefiniranimi vrednostmi.



Slika 4.10: Seznam temperaturnih senzorjev

Na grafu 4.10 lahko vidimo vrednosti izbranih senzorjev v odvisnosti od časa. Za lažji pregled je na voljo tudi legenda z opisi in barvami senzorjev. V primeru, da naprava podpira tudi večdotično interakcijo, jo lahko z gesto stisni tudi približamo v izbrano območje. Prav tako se lahko premikamo levo ali desno. Premikanje vertikalno pomeni, da se sprehajamo skozi vrednosti v času.

Poglavje 5

Sklepne ugotovitve

V diplomskem delu je predstavljen model avtomatizacije krmiljenja toplotne tehnike, za konkretni primer kopališča Laguna Ljubljana. Namen diplomske naloge je bil zajeti specifične zahteve in implementirati sistem, ki bo deloval optimalno v danih razmerah.

Pred samim načrtovanjem sistema smo že imeli točno določene zahteve, ki jih mora sistem upoštevati. Največja omejitev je bila prav s strani že kupljene strojne opreme; temperaturni senzorji in rele enote. Prav omejitev s strojno opremo nam je povzročala največ težav. Strojna oprema, kot so nadzorne enote za merjene temperature, niso uporabljale nobenega znanega protokola. Prav tako ni bilo nobene javne dokumentacije o delovanju naprave. Na voljo je bila samo skopa priložena programska oprema. To je pomenilo, da je bilo potrebno za veliko večino naprav naprej skrbno preučiti protokole. Pri tem smo uporabili posebna orodja in obratno inženirstvo. Kot se je kasneje izkazalo, je bil to najbolj zahteven del naloge.

Sistem je načrtovan tako, da se brez velikih sprememb lahko dodajajo gonilniki za nove naprave. Pri tem pa se koncept delovanja sistem ne spremeni. Avtomatizacijo krmiljenja in sam sistem bi ocenili kot uspešen, saj je v uporabi že slabo leto. Med zimo 2012/2013 pa se je zaradi optimizacije delovanja in razporejanja toplote poznalo tudi v manjših stroških energentov. V času, odkar se je sistem začel uporabljati, so se pokazale potrebe

po še kompleksnejši odločitveni logiki, ki bi omogočala zapletenejše pogoje. Naslednja faza je prav implementacija teh novih zahtev.

Literatura

- [1] (2013) "INSTAL.si spletna trgovina".
Dostopno na: http://www.instal.si/web_prospekti/558_Grundfos_UPS.pdf
- [2] PCsensor.com (2013) "Web Interface / Ethernet Thermometer, 4 channels thermometer(1W340A-D4)".
Dostopno na: http://www.pcsensor.com/index.php?_a=product&productId=9&product_id=9
- [3] Maxim Integrated "DS18B20 Programmable Resolution 1-Wire Digital Thermometer".
Dostopno na: <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [4] PCsensor.com (2013) "PCsensor- one outer sensor thermoemter , computer thermometer (TEMPer1)".
Dostopno na: http://www.pcsensor.com/index.php?_a=product&product_id=7
- [5] DENKOVI (2013) "Internet/Ethernet Eight Channel Relay Board". Dostopno na: <http://denkovi.com/productv/87/internet-ethernet-eight-channel-relay-board-ver-2.html>
- [6] Wikipedia "Android (operating system)".
Dostopno na: [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
- [7] Wikipedia "Microsoft Visual Studio".
Dostopno na: http://en.wikipedia.org/wiki/Microsoft_Visual_Studio

- [8] Wireshark (2013) "Wireshark User's Guide".
Dostopno na: <http://www.wireshark.org/download/docs/user-guide-a4.pdf>

- [9] Jonas Gehring (2013) "GraphView Library".
Dostopno na: <http://www.jjoe64.com/p/graphview-library.html>